



Ramp Up !:

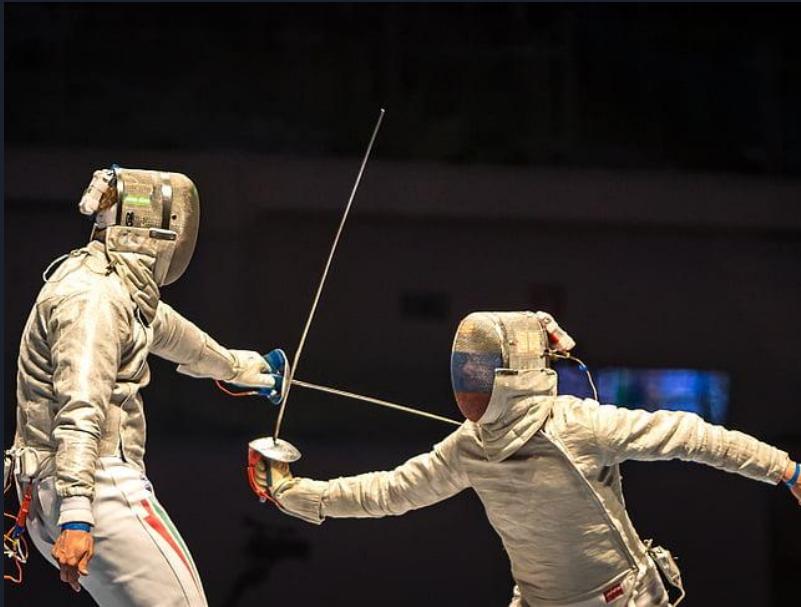
A Web App for Athletic Equipment Managers

February 23, 2021



Before we start . . .
A HUGE THANK YOU TO
EVERYONE <3!

Would not have pulled
through without all your
inspirations and camaraderie!





Our story

Why is it needed?

To provide a more efficient workflow by allowing equipment managers to enter, edit, delete, and locate orders and items quickly.

Is this just another general inventory app?

This app was made with the needs of an athletic department in mind by an active NCAA DIII equipment manager who wanted a more efficient workflow.

What is Ramp Up '!' ?

An logistical web app for daily operations in managing athletic equipment and gear.

Who benefits from this app?

Low budget athletic institutions, small sized, or non-tech savvy athletic departments

More Features to come...

Why I created this Web App

Exhausting too much resources:

Time

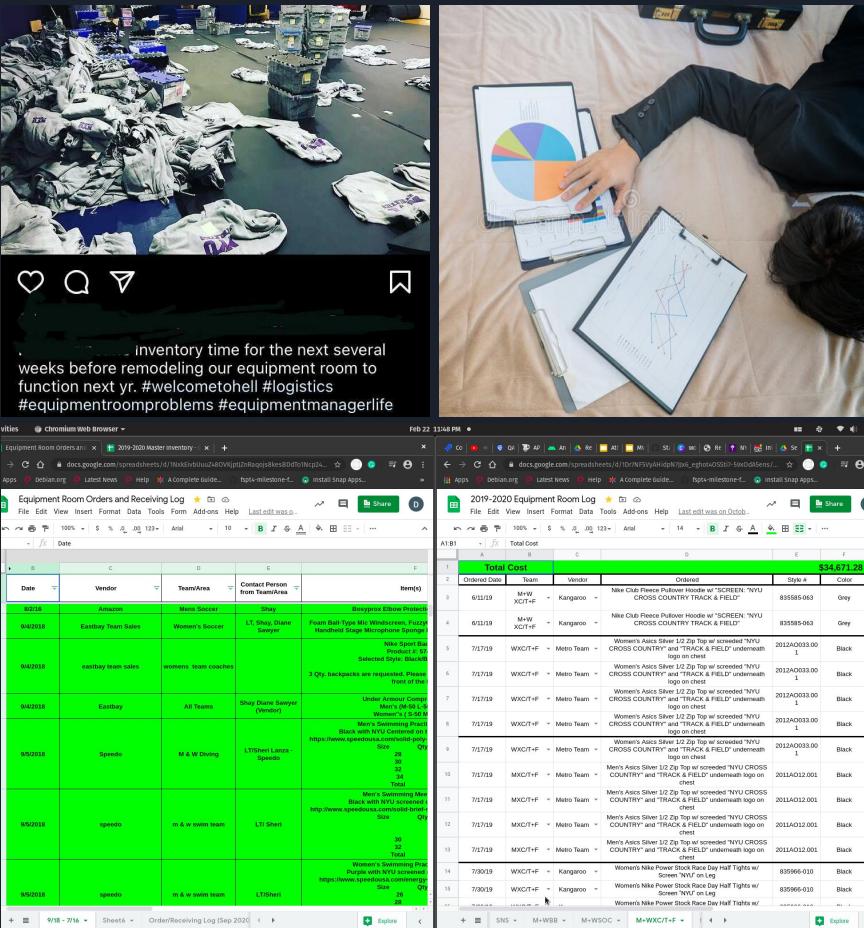
Recounting orders and inventory due to lack of tracking paper trail

Paperwork

Using dozens of separated documents in various softwares, locations, and/or permission restricted files

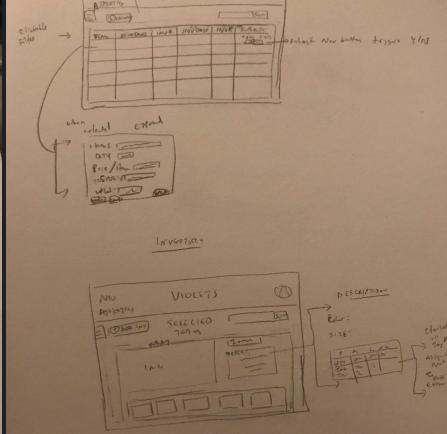
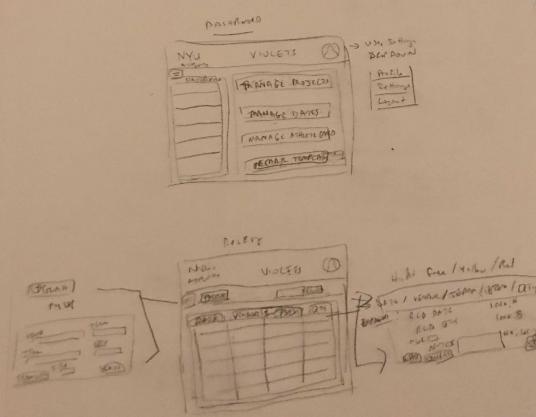
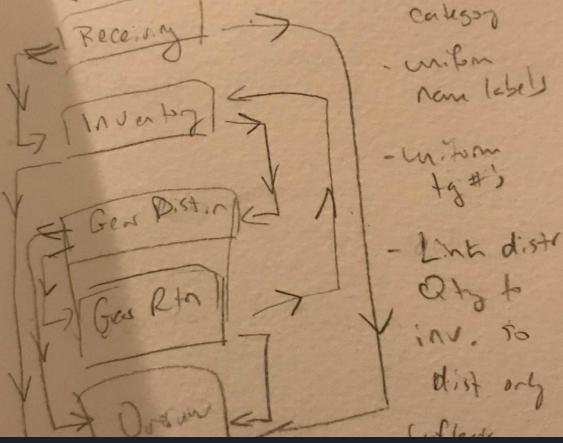
Efficiency

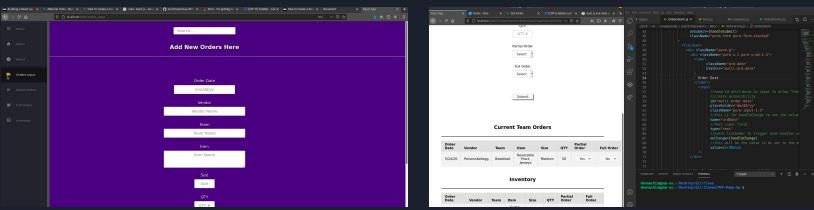
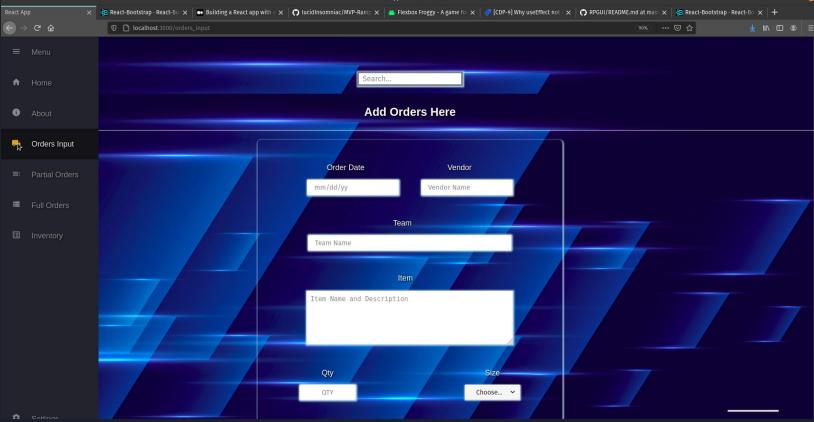
Repetitive and inaccurate recorded data being generated by different people simultaneous or different intervals, no version or unified control.



... The process ... Loading ...

How did the design process start?





Frontend Development

Progression of the web application from beginning to current.

```

node ./api/trapup
101 // In MySQL, "describe inventory"
102 router.post('/trapup', async (req, res) => {
103   const { ord_date, vendor, team, item, size, qty } = req.body
104   const { part_ord, full_ord, inv_id } = req.params
105   // Has to be in MySQL syntax
106   const sql = `INSERT INTO orders
107     (ord_date, vendor, team, item, size, qty, part_ord, full_ord, inv_id)
108   VALUES (
109     '$(ord_date)',
110     '$(vendor)',
111     '$(team)',
112     '$(item)',
113     '$(size)',
114     '$(qty)',
115     '$(part_ord)',
116     '$(full_ord)',
117     '$(inv_id)'
118   )`;
119   try {
120     await pool.query(sql);
121     res.json({ message: 'Success!' });
122   } catch (err) {
123     res.status(500).json({ error: err.message });
124   }
125 });
126
127 module.exports = router;
128
129 // Whenever we access a DB with "async" and "await", we need the "try"
  
```

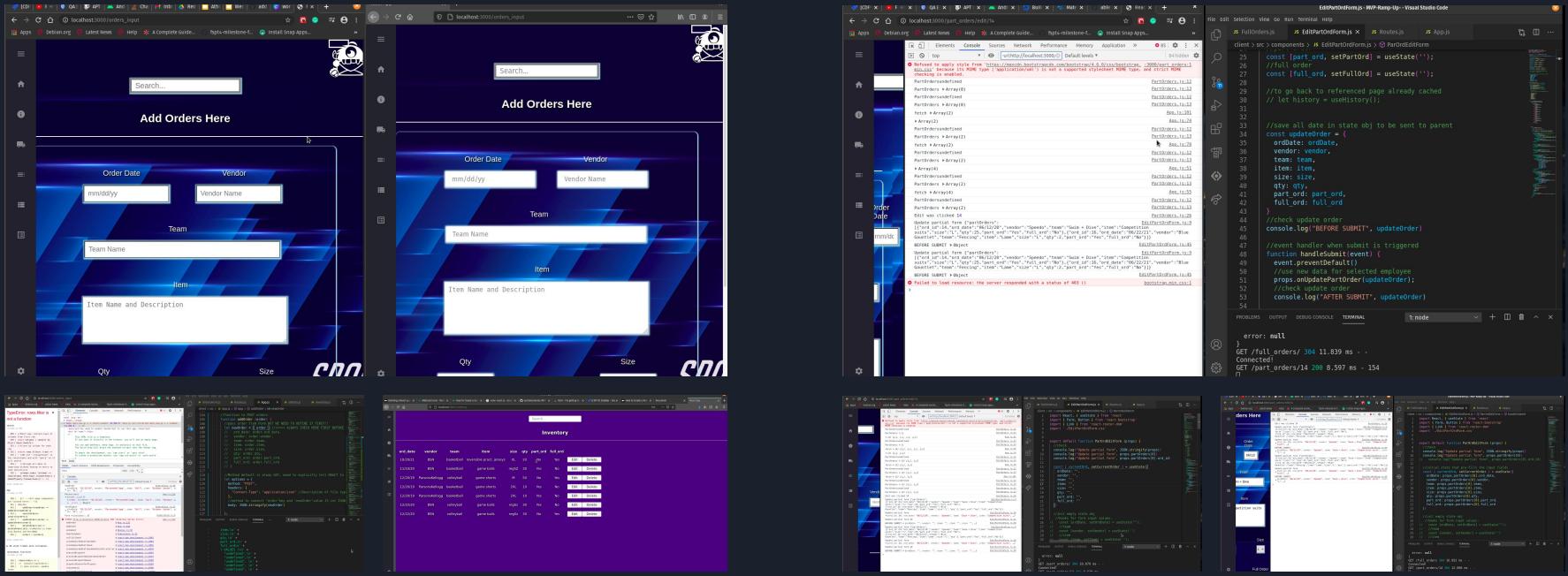
Body	Headers	Method	URL	Preview	Responses
<code>{ "ord_date": "12/28/18", "vendor": "ParsonsLogistics", "team": "Sales", "item": "Saw", "size": "M", "part_ord": "1", "full_ord": "1", "inv_id": "1" }</code>		POST	http://localhost:5000/api/trapup	Success!	201

Backend Development

Progression and testing of setting up MYSQL database and backend server with routes.

Technologies and Resources

Technologies	Tutorials and References	Misc...
<p>Frontend:</p> <ul style="list-style-type: none">- React JS- React-Bootstrap- React Router <p>Backend:</p> <ul style="list-style-type: none">- MySQL: Database- Express JS: server- Postman: testing backend API	<ul style="list-style-type: none">- CSS Tricks- Youtube- Google- FreeCodeCamp- Stack Overflow- Dev.to	<ul style="list-style-type: none">- Material UI Icons: Sidebar icons- Font Awesome: avatar icon mockup- Canva.com: Chart mockup- Jira: Bug and error tracking



Challenges

Some visuals of the issues encountered through creating the app.

Breadcrumb trails are a lifeline

After completing a segment or component, I console.log with comments to leave a breadcrumb trail to help debug.

How I was able to Stay on top Of bugs and errors

Using tools like Trello, GitHub Project, or Jira:

Create a new Log for each issue

This helped me so much instead of scrolling through my notes

Set priority on issue

If it was not a major issue, skip it and come back later to fix

Record details, and visuals

Write every step down that caused the issue, what the error message said, or side effects. Add any visuals and recordings. Dev Tools has a record that helps save to a file.

The screenshot shows a Jira project interface with several open issues. One issue, titled "Rows.filter not function on... Actually due to several reasons", is highlighted. The issue details page includes sections for Assignee (Donna Chin), Reporter (Donna Chin), Labels (None), Priority (Medium), and attachments (four screenshots of DevTools recordings). Below the attachments, there's a note: "CORRECTION: do not need to remove juice, only needed to redo yarn migrate and restart". Another issue, "Sidebar", is also visible. The bottom of the screen shows a list of all issues in the project, including their type, key, summary, assignee, reporter, priority, status, and creation date. The list includes items like "Search Bar", "Why useEffect not responding to DB", "Setting Diff Routes for same table", "ReactTable", "MySQL: Grouping data together by one category", "MySQL: Create table with Foreign Key", "AthleteImage image data exist but image not rendering properly", "Added scaffold and tried to push to git", "AthleteForm.js - Frontend Static page creation Only", and "Manager Access". Most issues have been resolved ("DONE") and were created between February 3rd and 26th, 2021.

Type	Key	Summary	Assignee	Reporter	P	Status	Created
Rows.filter not function on...	COP-12	Actually due to several reasons	Unassigned	Donna Chin	↑	DONE	Feb 26, 2021
Sidebar	COP-11		Unassigned	Donna Chin	↑	DONE	Feb 15, 2021
Search Bar	COP-10		Unassigned	Donna Chin	↑	DONE	Feb 10, 2021
Why useEffect not responding to DB	COP-9		Donna Chin	Donna Chin	↑	DONE	Feb 10, 2021
Setting Diff Routes for same table	COP-8		Unassigned	Donna Chin	↑	DONE	Feb 8, 2021
ReactTable	COP-7		Unassigned	Donna Chin	↑	DONE	Feb 8, 2021
MySQL: Grouping data together by one category	COP-6		Unassigned	Donna Chin	↑	DONE	Feb 4, 2021
MySQL: Create table with Foreign Key	COP-5		Donna Chin	Donna Chin	↑	DONE	Feb 4, 2021
AthleteImage image data exist but image not rendering properly	COP-4		Unassigned	Donna Chin	↓	IN PROGRESS	Feb 4, 2021
Added scaffold and tried to push to git	COP-3		Unassigned	Donna Chin	↑	DONE	Feb 4, 2021
AthleteForm.js - Frontend Static page creation Only	COP-2		Donna Chin	Donna Chin	↑	DONE	Feb 3, 2021
Manager Access	COP-1		Donna Chin	Donna Chin	↑	DONE	Feb 2, 2021

... The Aftermath ... and Reflections ...

Most Difficult	Lessons Learned	Most Proud Moments
<p>Frontend:</p> <ul style="list-style-type: none">- Front-middle connection- React-Bootstrap- React Router <p>Backend:</p> <ul style="list-style-type: none">- MySQL: Database- DB Schema- Express JS: server	<ul style="list-style-type: none">- Keep a proj diary- Log every issue- Update fixed issues- Comment in code, anything unclear- Naming properly to variables and functions- Check cross browser rendering- Step away when needed- Sleep ...- Try out tutorials or code in practice before on your code	<ul style="list-style-type: none">- Responsive sliding sidebar in pure CSS- Search bar Function- Forms for the most part are functional- Getting order by Id and auto-populate in Edit Form- Connection frontend to backend- Testing and logging everything



Future Features

Create user accounts

Allow users to log in and use app.

Re-design Database Schema

As the app grows with more data and functions, the database needs to be routed correctly to pass requested data

Solve Edit Button issue

Edit button function to push updated data is not responsive.
Isolated issue in Routes.js and page for Edit Form.

Login and Registration Pages

Create login system for user accounts

More Features to come...

- Security
- Work with GoogleAPIs
- QR and Barcode API
- Add Confluence internal FAQ
- And more...